Two SVDs produce more focal deep learning representations

Hinrich Schütze
Center for Information and Language Processing
University of Munich, Germany
hs999@ifnlp.org

Christian Scheible
Institute for NLP
University of Stuttgart, Germany
scheibcn@ims.uni-stuttgart.de

Abstract

A key characteristic of work on deep learning and neural networks in general is that it relies on representations of the input that support generalization, robust inference, domain adaptation and other desirable functionalities. Much recent progress in the field has focused on efficient and effective methods for computing representations. In this paper, we propose an alternative method that is more efficient than prior work and produces representations that have a property we call *focality* – a property we hypothesize to be important for neural network representations. The method consists of a simple application of two consecutive SVDs and is inspired by (Anandkumar et al., 2012).

In this paper, we propose to generate representations for deep learning by two consecutive applications of singular value decomposition (SVD). In a setup inspired by (Anandkumar et al., 2012), the first SVD is intended for denoising. The second SVD rotates the representation to increase what we call *focality*. In this initial study, we do not evaluate the representations in an application. Instead we employ diagnostic measures that may be useful in their own right to evaluate the quality of representations independent of an application.

We use the following terminology. SVD^1 (resp. SVD^2) refers to the method using one (resp. two) applications of SVD; 1LAYER (resp. 2LAYER) corresponds to a single-hidden-layer (resp. two-hidden-layer) architecture.

In Section 1, we introduce the two methods SVD^1 and SVD^2 and show that SVD^2 generates better (in a sense to be defined below) representations than SVD^1. In Section 2, we compare 1LAYER and 2LAYER SVD^2 representations and show that 2LAYER representations are better. Section 3 discusses the results.

1 SVD^1 vs. SVD^2

Given a base representation of n objects in $\mathcal{R}^d$, we first compute the first k dimensions of an SVD on the corresponding $n \times d$ matrix C. $C_k = USV^T$ (where C_k is the rank-k approximation of C). We then use US to represent each object as a k-dimensional vector. Each vector is normalized to unit length because our representations are count vectors where the absolute magnitude of a count contains little useful information – what is important is the relative differences between the counts of different dimensions. This is the representation SVD^1. It is motivated by standard arguments for representations produced by dimensionality reduction: compactness and noise reduction. Denoising is also the motivation for the first SVD in the method proposed by Anandkumar et al. (2012).

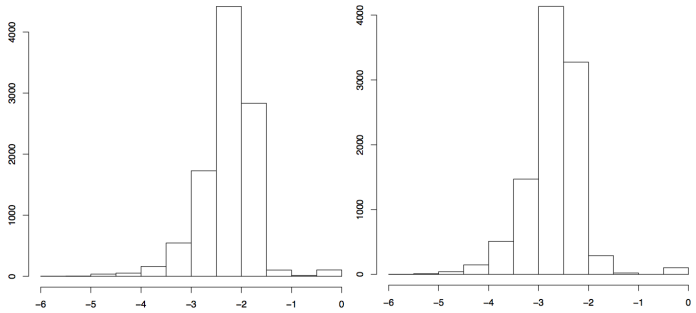

Figure 1: Histograms of $\log_{10}|c|$ of correlation coefficients of SVD^1 (left) and SVD^2 (right).

We then perform a second SVD on the resulting matrix C' of dimensionality $n \times k$.[1] $C' = U'S'V'^T$ (full-rank, no dimensionality reduction). We again use $U'S'$ to represent each object as a k-dimensional vector. Each vector is normalized to unit length. This is the representation SVD^2.

SVD^2 is intended to be a rotation of SVD^1 that is more "focal" in the following sense. Consider a classification problem f over a k-dimensional representation space R. Let $M(f, R)$ be the size k' of the smallest subset of the dimensions that support an accuracy above a threshold for f. Then a representation R is more focal than R' if $M(f, R) < M(f, R')$. The intuition is that good deep learning representations have semantically interpretable hidden units that contribute input to a decision that is to some extent independent of other hidden units. We want the second SVD to rotate the representation into a "more focal" direction.

The role of the second SVD is somewhat analogous to that of the second SVD in the approach of Anandkumar et al. (2012), where the goal also is to find a representation that reveals the underlying structure of the data set.

Experimental setup. We use a corpus of movie review sentences (Pang and Lee, 2004). Following Schütze (1995), we first compute a left vector and a right vector for each word. The dimensionality of the vectors is 250. Entry i for the left (right) vector of word w is the number of times that the word with frequency rank i occurred immediately to the left (right) of w. Vectors are then tf-idf weighted and length-normalized. We randomly select 100,000 unique trigrams from the corpus, e.g., "tangled feelings of" or "as it pushes". Each trigram is represented as the concatentation of six vectors, the left and right vectors of the three words. This defines a matrix of dimensionality $n \times d$ ($n = 100000$, $d = 1500$). We then compute SVD^1 and SVD^2 on this matrix for $k = 100$.

Analysis of correlation coefficients. Figure 1 shows histograms of the 10,000 correlation coefficients of SVD^1 (left) and SVD^2 (right). Each correlation coefficient is the correlation of two columns in the corresponding 100000×100 matrix and is transformed using the function $f(c) = \log_{10}|c|$ to produce a histogram useful for the analysis. The histogram of SVD^2 is shifted by about 0.5 to the left. This is a factor of $10^{0.5} \approx 3$. Thus, SVD^2 dimensions have correlations that are only a third as large as SVD^1 correlations on average.

We take this to indicate that SVD^2 representations are more focal than SVD^1 representations because the distribution of correlation coefficients would change the way it changes from SVD^2 to SVD^1 if we took a focal representation (in the most extreme case one where each dimension by itself supported a decision) and rotated it.

Discrimination task. We randomly selected 200 words in the frequency range $[25, 250]$ from the corpus; and randomly arranged them into 100 pairs. For each pair, we first retrieved the SVD^1 and SVD^2 representations of all triples from the set of 100,000 in which one of the two words was the central word. Then we determined for each dimension i of the 100 dimensions (for both SVD^1 and SVD^2) the optimal discrimination value θ by exhaustive search; that is, we determined the threshold θ for which the accuracy of the classifer $\vec{v}_i > \theta$ (or $\vec{v}_i < \theta$) was greatest – where the discrimination

[1] Since an SVD is a linear operation, it may at first seem questionable to perform two SVDs in sequence. Indeed, the SVD decomposition of unreduced US consists of the triple of matrices <identity matrix, S, U> – i.e., a "full-rank" second SVD doesn't do anything. However, for dimensionality reduction ($k < d$) and length-normalized vectors, SVD^2 is in general not a linear function of the input representation.

task was to distinguish triples that had one word vs the other as their central word. Finally, of the 100 discrimination accuracies we chose the largest one for this word pair.

On this discrimination task, SVD^2 was better than SVD^1 55 times, the two were equal 15 times and SVD^2 was worse 30 times. On average, discrimination accuracy of SVD^2 was 0.7% better than that of SVD^1. This is evidence that SVD^2 is better for this discrimination task than SVD^1.

This indicates again that SVD^2 representations are more focal than SVD^1 representations: each dimension is more likely to provide crucial information by itself as opposed to only being useful in conjunction with other dimensions.

2 1LAYER vs. 2LAYER

We compare two representations of a word trigram: (i) the 1LAYER representation from Section 1 and (ii) a 2LAYER representation that goes through two rounds of autoencoding, which is a deep learning representation in the sense that layer 2 represents more general and higher-level properties of the input than layer 1.

To create 2LAYER representations, we first create a vector for each of the 20701 word types occurring in the corpus. This vector is the concatenation of its left vector and its right vector. The resulting 20701 $\times$ 500 matrix is the input representation to SVD^1. We again set $k = 100$. A trigram is then represented as the concatenation of three of these 100-dimensional vectors. We apply the SVD^2 construction algorithm to the resulting 100000 $\times$ 300 matrix and truncate to $k = 100$.

We now have – for each trigram – two SVD^2 representations, the 1LAYER representation from Section 1 and the 2LAYER representation we just described. We compare these two trigram representations, again using the task from Section 1: discrimination of the 100 pairs of words.

2LAYER is better than 1LAYER 64 times on this task, the same in 18 cases and worse in 18 cases. This is statistically significant ($p < .01$, binomial test) evidence that 2LAYER SVD^2 representations are more focal than 1LAYER SVD^2 representations.

3 Discussion

One advantage of focal representations is that many classifiers cannot handle conjunctions of several features unless they are explicitly defined as separate features. Compare two representations $\vec{x}$ and $\vec{x}'$ where $\vec{x}'$ is a rotation of $\vec{x}$ (as it might be obtained by an SVD). Since one vector is a rotation of the other, they contain exactly the same information. However, if (i) an individual "hidden unit" of the rotated vector $\vec{x}'$ can directly be interpreted as "is verb" (or a similar property like "is adjective" or "takes NP argument") and (ii) the same feature requires a conjunction of several hidden units for $\vec{x}$, then the rotated representation is superior for many upstream statistical classifiers.

Focal representations can be argued to be closer to biological reality than broadly distributed representations (Thorpe, 2010); and they have the nice property that they become categorical in the limit. Thus, they include categorical representations as a special case.

A final advantage of focal representations is that in some convolutional architectures the input to the top-layer statistical classifier consists of maxima over HU (hidden unit) activations. E.g., one way to classify a sentence as having positive/negative sentiment is to slide a neural network whose input is a window of k words (e.g., $k = 4$) over it and to represent each window of k words as a vector of HU activations produced by the network. In a focal representation, the hidden units are more likely to have clear semantics like "the window contains a positive sentiment word". In this type of scenario, taking the maximum of activations over the $n - k + 1$ sliding windows of a sentence of length n results in hidden units with interpretable semantics like "the activation of the positive-sentiment HU of the window with the highest activation for this HU". These maximum values are then a good basis for sentiment classification of the sentence as a whole.

If a direct comparison of SVD^2 with traditional deep learning (Hinton et al., 2006) were to show that the two approaches are equally powerful, then SVD^2 would be an interesting alternative to deep learning initialization methods currently used since SVD is efficient and a simple and well understood formalism.

References

Animashree Anandkumar, Dean P. Foster, Daniel Hsu, Sham M. Kakade, and Yi-Kai Liu. 2012. Two svds suffice: Spectral decompositions for probabilistic topic modeling and latent dirichlet allocation. *CoRR*, abs/1204.6703.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of ACL*.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 141–148.

Simon Thorpe. 2010. Grandmother cells and distributed representations. In Nikolaus Kriegeskorte and Gabriel Kreiman, editors, *Understanding visual population codes. Toward a common multivariate framework for cell recording and functional imaging*. MIT Press.